

I linguaggi di programmazione

Cos'è l'Informatica

Informatica = Informazione (dati + istruzioni) + Automazione.

Cos'è il Computer

Insieme organizzato di componenti, in grado di eseguire una successione di istruzioni finalizzate a elaborare l'informazione. È in grado di eseguire operazioni relativamente semplici ad alta velocità.

Cos'è un algoritmo

Sequenza **finita** e **ordinata** di passi/operazioni che portano alla realizzazione di uno specifico compito. L'algoritmo di un'operazione complessa può essere scomposto in una sequenza di istruzioni più semplici.

Esempi: calcoli matematici, massimo comune divisore, istruzioni di un elettrodomestico. Prelevamento Bancomat.

N.B. Un computer è un esecutore di algoritmi

Proprietà di un algoritmo: **correttezza** (deve giungere alla soluzione del dato problema) ed **efficienza** (dare la soluzione nel modo più veloce, utilizzando la minima quantità di risorse fisiche).

Metodi di rappresentazione di un algoritmo:

- Linguaggio naturale
- Diagramma a blocchi
- Pseudo codice
- Linguaggio di programmazione

Cos'è un programma

Insieme di uno o più algoritmi scritti in un determinato linguaggio di programmazione. Il processore del computer esegue i programmi passo-passo in modo preciso e veloce.

Gli algoritmi sono parametrici

- Producono un risultato che dipende da un insieme di dati di partenza
- Descrivono la soluzione non di un singolo problema, ma di una intera classe di problemi strutturalmente equivalenti

Esempi:

- l'algoritmo per la moltiplicazione di due numeri specifica come effettuare il prodotto di tutte le possibili coppie di numeri
- l'algoritmo per la ricerca di un libro nello schedario della biblioteca vale per tutti i possibili libri

Le istruzioni dell'algoritmo fanno riferimento a **variabili**, il cui valore non è fissato a priori ma cambia a seconda della situazione elaborativa in cui il processore del computer si trova.

Cos'è una variabile

È un dato, formato da un'etichetta e un valore, mantenuta all'interno della memoria RAM

Uso delle variabili

- All'interno di espressioni (l'esecutore usa il valore contenuto nelle variabili per calcolare il risultato dell'espressione, per esempio $op1 + op2 \times op3$ oppure $op1 / op2 - op3$, ...)
- In istruzioni di assegnamento (introdurre nel contenitore identificato dal nome della variabile il valore specificato a destra dell'assegnamento, per esempio $r = 35$ (assegna 35 alla variabile il cui nome è r), $pi = 3,14$, ...)

- In istruzioni di assegnamento combinate con espressioni (assegna a una variabile il risultato ottenuto dalla valutazione di un'espressione, per esempio in " $\text{circ} = 2 \times r \times \text{pi}$ " il risultato dell'espressione $2 \times r \times \text{pi}$ viene calcolato utilizzando i valori contenuti nelle variabili r e pi e il risultato viene poi assegnato alla variabile circ ; la stessa variabile può comparire in entrambi i lati dell'istruzione di assegnamento, per esempio in " $k = k + 1$ " il valore contenuto in k viene utilizzato per trovare il valore dell'espressione $k + 1$ che viene memorizzato come nuovo valore di k .)

Assegnazione di valori a variabili

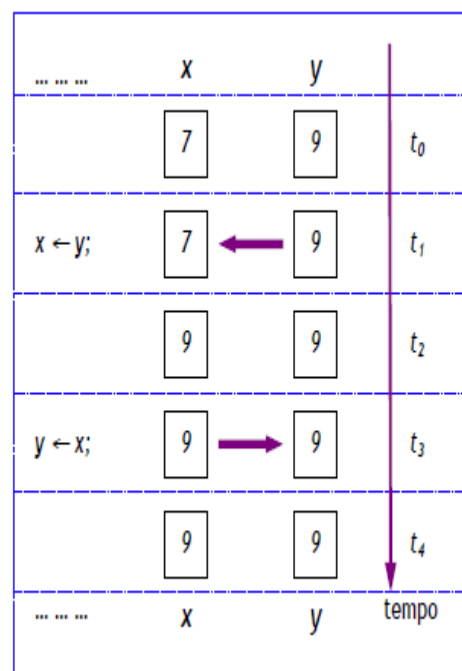
Il valore assegnato a una variabile si sostituisce a quello che era presente in precedenza, il vecchio valore non potrà più essere recuperato.

Esempio: si ipotizzi di voler scambiare i valori contenuti in due variabili x e y .

Soluzione proposta: doppio assegnamento del tipo

$x \leftarrow y$
 $y \leftarrow x$

Per indicare che il valore di y deve essere copiato in x e che, nello stesso tempo, il valore di x sia trasferito in y . Le istruzioni però vengono eseguite in sequenza! Quindi l'assegnamento $x \leftarrow y$ viene completato prima di iniziare $y \leftarrow x$.



...	tmp	x	y	
	??	7	9	t_0
$tmp \leftarrow x;$	7	7	9	t_1
	7	7	9	t_2
$x \leftarrow y;$	7	7	9	t_3
	7	9	9	t_4
$y \leftarrow tmp;$	7	9	9	t_5
	7	9	7	t_6
...	tmp	x	y	tempo

Soluzione corretta: uso di una variabile aggiuntiva (**tmp**), come strumento di memorizzazione temporanea ("buffer") del valore originariamente contenuto in **x**

$tmp \leftarrow x$

$x \leftarrow y$

$y \leftarrow tmp$

In questo modo lo scambio avviene senza perdere i valori originali.

Dati e Istruzioni

• Tipi di dati

- Numeri naturali o interi o reali (1, -2, 0.34)
- Caratteri alfanumerici (A, B, a, c, 1, 2, -, |, ...)
- N.B. un insieme ordinato di caratteri alfanumerici viene definita stringa.
- Dati logici o booleani (true, false)
- Array o vettore di n elementi ({1, 2, 3})

• Istruzioni

- Operazioni di Input/Output (es. leggi, scrivi)
- Operazioni Aritmetico-logiche (es. somma = $A + B$)
- Strutture di controllo (es. SE, RIPETI)

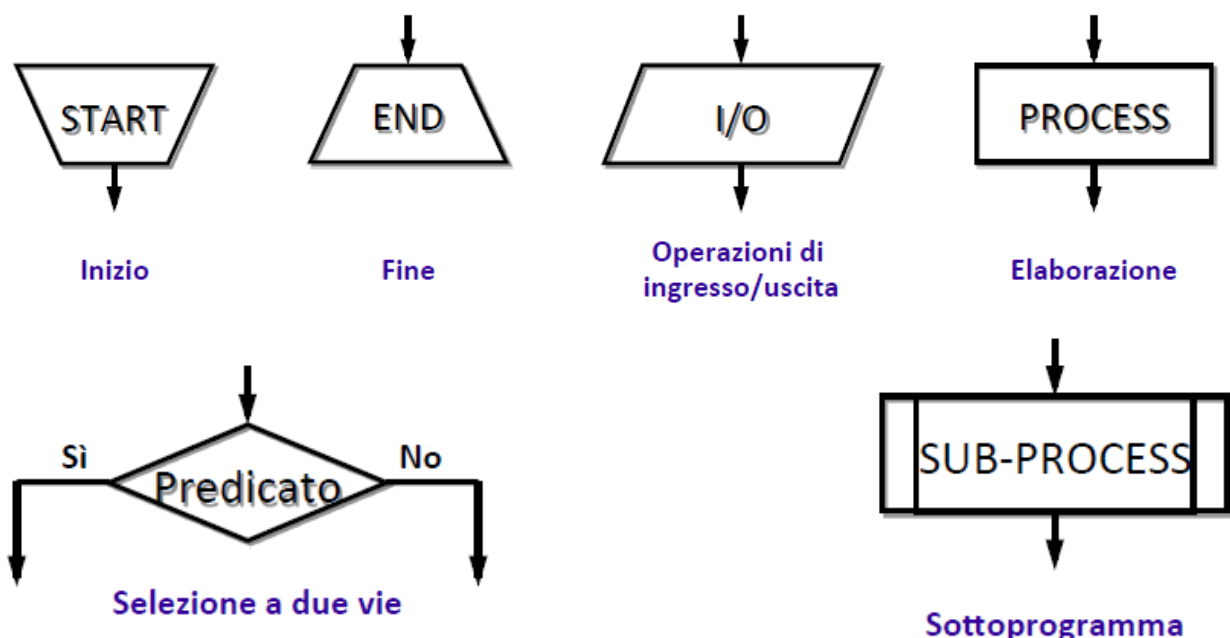
• Struttura dati di tipo

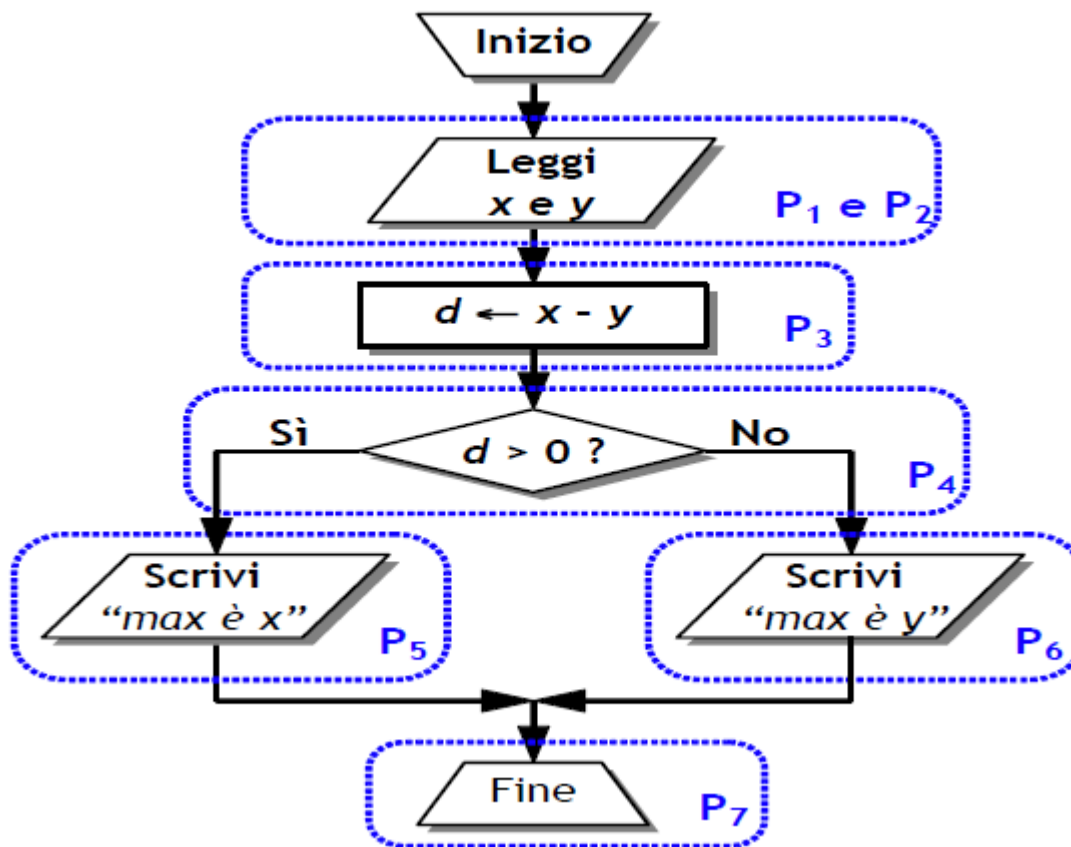
- elementari (interi, alfanumerici, booleani, ...)
- strutturati (array, matrici, ...)

Esempio di programma: Determinare il maggiore tra due numeri

P1 leggi un valore dall'esterno e assegnalo alla variabile x ;
P2 leggi un secondo valore dall'esterno e assegnalo alla variabile y ;
P3 calcola la differenza d fra x e y , cioè esegui $d \leftarrow x - y$;
P4 valuta se d è positivo: in caso *affermativo* prosegui con il passo P5, altrimenti (in caso *negativo*) salta al passo P7;
P5 scrivi "il numero maggiore è " seguito dal valore di x ;
P6 salta al passo P11;
P7 valuta se d è nullo: in caso *affermativo* prosegui con il passo P8, altrimenti (in caso *negativo*) salta al passo P10;
P8 scrivi "i due numeri sono uguali";
P9 salta al passo P11;
P10 scrivi "il numero maggiore è " seguito dal valore di y ;
P11 termina l'esecuzione.

Struttura dei diagrammi di flusso





Leggi alfa, beta;

prodotto \leftarrow 0;

Finché alfa > 0 ripeti

 prodotto \leftarrow prodotto + beta;

 alfa \leftarrow alfa - 1;

stampa prodotto;